

REMARKS

Claims 1-6, 8-24, and 26-32 are pending in the present application; claims 7 and 25 have been cancelled; claims 33-36 have been added. The Examiner rejected: claims 19-32 under 35 U.S.C. § 101 as being directed to non-statutory subject matter; claims 1-3, 15-17, and 19-21 under 35 U.S.C. § 102(e) as being anticipated by *Surasinghe* (pat. app. 2004/0194069); claims 4, 18, and 22 under 35 U.S.C. § 103(a) as being unpatentable over *Surasinghe* in view of *Ward* (U.S. Pat. No. 5,103,421); claims 5-10 and 23-28 under § 25 U.S.C. § 103(a) as being unpatentable over *Surasinghe* in view of *Viswanath* (pat. app. 2004/0019670); claims 11-14 and 29-32 as being unpatentable over *Surasinghe* in view of *Viswanath* and further in view of *Cronin* (U.S. Pat. No. 6,772,396).

Independent claims 1, 15, and 19 generally call for, among other things, “generating at least one second level of the configuration items based on the modification to the at least one first level.” Independent claims 1, 15, and 19 are generally directed to a software system, where a second level of configuration items is generated based on changes made to a first level of configuration items. Such a software system removes the necessity to rewrite the second level configuration items when changes are made to the first level configuration items. As a result, the ability to communicate and/or transfer information between the levels of configurations or between programs and/or systems utilizing the configuration levels is maintained. *See, e.g.*, p. 19, lines 2-7 [¶ 72] (“In the course of modifying configuration items in response to GUI commands, changes are made to high-level configuration items. These changes include operations such as additions, updates, and deletions. The changes must be reflected in the corresponding low-level configuration XML, which in turn can then be

exported directly to the OSS 318, customer management database 226, and/or convergent billing database 220.”).

Applicant respectfully traverses the Examiner’s assertion that independent claims 1, 15, and 19 are anticipated by *Surasinghe* since *Surasinghe* fails to disclose a system which generates new configuration items at one level based on the changes made to configuration items at another level. Instead, *Surasinghe* discloses a system in which changes to rules used by a single application program are converted to external executable files (.dll files) that are called by the application program when needed. While the *Surasinghe* system creates executable routines that may be used by the application program, the *Surasinghe* system fails to teach, disclose, or suggest a system that generates a second set of configuration items based on changes made to a first set of configuration items that are used by other support systems and that differs from the original set of the configuration items prior to the changes. Moreover, *Surasinghe* fails to teach, disclose, or suggest a system having a second level of configuration (*i.e.*, other support systems) or making changes to the configuration items of a second level based on the changes made to the first level.

Independent claims 5 and 23 generally call for, among other things:

modifying the one or more configuration items, wherein the modifying includes modifying high-level representations of the one or more configuration items associated with at least one first system; generating low-level representations of the one or more configuration items associated with at least one second system.

Independent claims 5 and 23 are generally directed to a software system, where low-level representations of configuration items associated with one or more additional systems are generated based on the changes made to high-level representations of configurations items associated with another system.

Applicant respectfully traverses the Examiner's assertion that independent claims 5 and 23 are made obvious by *Surasinghe* in view of *Viswanath*. In particular, *Surasinghe* fails to disclose a system that generates new representations of configuration items at one level based on the changes made to representations of configuration items at another level. As set forth above, *Surasinghe* instead discloses a system in which changes to rules used by a single application program are converted to external executable files (.dll files) that are called by the application program when needed. If the changes to the business rules also require changes to the database that may include the modification, addition, or deletion of new fields or sub-fields, the *Surasinghe* system includes a dynamic database so that these changes are made to the database as well. While the *Surasinghe* system makes changes to the business rules and supporting database that may be used by the application program, the *Surasinghe* system fails to teach, disclose, or suggest a system that generates a second set of configuration item representations based on changes made to a first set of configuration items. Moreover, *Surasinghe* fails to teach, disclose, or suggest a system having additional systems, additional databases, and a second level of configuration items used by other support systems and their databases, or making changes to the configuration items of a second level based on the changes made to the first level.

Viswanath also fails to disclose a system that generates new representations of configuration items at one level based on changes made to representations of configuration items at another level. Instead, *Viswanath* ostensibly discloses a system that stores server-level and sub-level configuration and is capable of displaying the configuration in a hierarchical model. The system described in *Viswanath* appears to generate "beans" that correspond to each element of the configuration, whether on the "server" level or another level, that allow

each configuration element to be modified using the corresponding bean. Although the beans associated with each configuration element disclosed in *Viswanath* appear to be automatically created, they do not, however, remove the requirement to make changes to configuration representations (manually, either to the element directly or via the corresponding bean) when changes are made to the configuration representations at another level. While the *Viswanath* system allows a user to alter configuration elements via the corresponding beans, it fails to teach, disclose, or suggest a system that either generates configuration representations or changes configuration representations at one level when changes are made to other configuration representations at another level.

Independent claims 11 and 29 generally call for, among other things,

creating, modifying, or deleting high-level configuration items associated with a second system based on the requested change to the first set of low-level configuration items; generating a second set of low-level configuration items based on the changes made to the high-level configuration items;

Independent claims 11 and 29 are generally directed to a software system, where changes to low-level configuration items associated with one or more additional support systems requested by a user are used to create, modify, or delete high-level configuration items associated with another system. A new set of low-level configuration items is then generated based on the changes made to high-level configurations items.

Applicant respectfully traverses the Examiner's assertion that independent claims 11 and 29 are made obvious by *Surasinghe* in view of *Viswanath* and in further view of *Cronin*. *Surasinghe* fails to disclose a system which generates new configuration items at one level based on the changes made to configuration items at another level. Instead, *Surasinghe* discloses a system in which changes to rules used by a single application program are

converted to external executable files (.dll files) that are called by the application program when needed. While the *Surasinghe* system creates executable routines that may be used by the application program, the *Surasinghe* system fails to teach, disclose, or suggest a system that generates a second set of configuration items when changes are made to another level of configuration. Moreover, *Surasinghe* fails to teach, disclose, or suggest a system having a second level of configuration (*i.e.*, other support systems) or changing configuration items of a second level based on the changes made to a first level.

Likewise, *Viswanath* fails to disclose a system which generates new configuration items at one level based on the changes made to configuration items at another level. Instead, *Viswanath* ostensibly discloses a system that stores server-level and sub-level configuration and displays the configuration in a hierarchical model. The system described in *Viswanath* appears to generate “beans” that correspond to each element of the configuration, whether on the “server” level or another level, that allow each configuration element to be modified using the corresponding bean. Although the beans associated with each configuration element disclosed in *Viswanath* appear to be automatically created, they do not, however, remove the requirement to make changes to configuration items (manually, either to the element directly or via the corresponding bean) when changes are made to the configuration items at another level. While the *Viswanath* system allows a user to alter configuration elements via the corresponding beans, it fails to teach, disclose, or suggest a system that either generates configuration items or changes configuration items at one level when changes are made to another level of configuration items.

Finally, *Cronin* fails to rectify the deficiencies found in *Surasinghe* or *Viswanath*. *Cronin* fails to disclose a system which generates new configuration items at one level based on

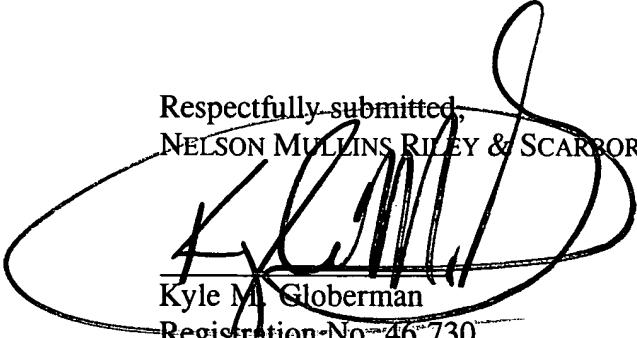
the changes made to configuration items at another level. Instead, *Cronin* is directed to a system that applies aesthetic formatting to information hosted by one website to match the appearance of other websites. In other words, *Cronin* discloses a system that retrieves information from one website, formats the information so it appears to match the visual layout of a first target website, and then is capable of applying additional formats so that the information may then match the visual layout of a second target website. The system disclosed in *Cronin* deals with formatting the output or display of a program and merely generates additional copies of the output. *Cronin* fails to teach, disclose, or suggest generating new configuration items used by one system based on requested changes to configuration items used by another system.

The Examiner rejected claims 19-32 under 35 U.S.C. § 101 as being directed to non-statutory subject matter requiring that “the software must be computer-readable.” Claims 19-32 have been amended to include “computer-readable.”

For the reasons stated above, independent claims 1, 15, and 19 are not anticipated by *Surasinghe*; independent claims 5 and 23 are not made obvious by *Surasinghe* in view of *Viswanath*; and independent claims 11 and 29 are not made obvious by *Surasinghe* in view of *Viswanath* in further view of *Cronin*. Additionally, claims 19-21 are directed to statutory subject matter. Dependent claims 2-4, 6-10, 12-14, 16-18, 20-22, 24-28, and 30-36 directly or indirectly depend from independent claims 1, 5, 11, 15, 19, 23, and 29. These dependent claims recite further limitations and are allowable in their respective combinations. Favorable action by the Examiner and withdrawal of the cited rejections is respectfully requested. The Examiner is invited to call the undersigned in an effort to discuss and resolve any remaining

issues. Please charge any additional fees or credit any overpayment to Deposit Account No. 50-1196.

Respectfully submitted,
NELSON MULLINS RILEY & SCARBOROUGH, LLP.


Kyle M. Globerman
Registration No. 46,730
1320 Main Street
Columbia, SC 29201
Office: (404) 817-6204
Fax: (803) 255-9831